LAMPIRAN A

KODE PROGRAM

```
%-------------------------------------------------------------------------------------
% create_gabor.m
close all;
clear all;
clc;

G = cell(5,8);
for s = 1:5
   for j = 1:8
      G{s,j}=zeros(32,32);
   end
end
for s = 1:5
   for j = 1:8
      G{s,9-j} = gabor([32 32],(s-1),j-1,4*pi/5,sqrt(2),3*pi/2);
   end
end

%figure;
for s = 1:5
   for j = 1:8
%       subplot(5,8,(s-1)*8+j);
%       imshow(real(G{s,j}),[]);
   end
end

for s = 1:5
   for j = 1:8
      G{s,j}=fft2(G{s,j});
   end
end
save gabor G
```

```
%-------------------------------------------------------------------------------------------
% gabor.m
function Psi = gabor (w,nu,mu,Kmax,f,sig)

% w  : Window [128 128]
% nu : Scale [0 ...4];
% mu : Orientation [0...7]
% kmax = pi/2
% f = sqrt(2)
% sig = 2*pi

m = w(1);
n = w(2);
K = Kmax/f^nu * exp(i*mu*pi/8);
Kreal = real(K);
Kimag = imag(K);
NK = Kreal^2+Kimag^2;
Psi = zeros(m,n);
for x = 1:m
   for y = 1:n
      Z = [x-m/2;y-n/2];
      Psi(x,y) = (sig^(-2))*exp((-.5)*NK*(Z(1)^2+Z(2)^2)/(sig^2))*...
            (exp(i*[Kreal Kimag]*Z)-exp(-(sig^2)/2));
   end
end

%-------------------------------------------------------------------------------------------
% createffnn.m
net = network;

net.numInputs = 1;
```

```
net.numLayers = 2;

net.biasConnect = [1;1];

net.inputConnect = [1 ;...
                    0 ];

net.layerConnect = [0 0 ;...
                    1 0 ];

net.outputConnect = [0 1];
net.targetConnect = [0 1];

netInputs = ones (2160,2);
netInputs (1:2160,1)= -1;
net.inputs{1}.range = netInputs;

net.layers{1}.size = 100;
net.layers{2}.size = 1;

net.layers{1:2}.transferFcn = 'tansig';
net.layers{1:2}.initFcn = 'initnw';

net.initFcn = 'initlay';
net.performFcn = 'msereg';
net.trainFcn = 'trainscg';

net = init(net)
save net net
```

```
%-------------------------------------------------------------------------------------------
% im2vec.m

function IMVECTOR = im2vec (W27x18)

load gabor;
W27x18 = adapthisteq(W27x18,'Numtiles',[8 3]);
Features135x144 = cell(5,8);
for s = 1:5
   for j = 1:8
      Features135x144{s,j} = ifft2(G{s,j}.*fft2(double(W27x18),32,32),27,18);
   end
end
Features135x144 = abs(cell2mat(Features135x144));
Features135x144 (3:3:end,:)=[];
Features135x144 (2:2:end,:)=[];
Features135x144 (:,3:3:end)=[];
Features135x144 (:,2:2:end)=[];
Features45x48 = premnmx(Features135x144);
IMVECTOR = reshape (Features45x48,[2160 1]);

%-------------------------------------------------------------------------------------------
% loadimages.m

function IMGDB = loadimages
%-------------------------------------------------------------------------------------------
face_folder = 'face\';  %Lokasi gambar wajah
non_face_folder = 'non-face\'; %Lokasi gambar non-wajah
file_ext = '.png';
out_max = 0.9;  % Output yang diinginkan untuk mendeteksi sebuah wajah
out_min = -0.9; % Output yang diinginkan untuk mendeteksi Non-wajah
%-------------------------------------------------------------------------------------------
```

```
 if exist('imgdb.mat','file')
    load imgdb;
else
    IMGDB = cell (3,[]);
end
fprintf ('Loading Faces ');
folder_content = dir ([face_folder,'*',file_ext]);
nface = size (folder_content,1);
for k=1:nface
    string = [face_folder,folder_content(k,1).name];
    image = imread(string);
    [m n] = size(image);
    if (m~=27 || n~=18)
        continue;
    end
    f=0;
    for i=1:length(IMGDB)
        if strcmp(IMGDB{1,i},string)
            f=1;
        end
    end
    if f==1
        continue;
    end
    fprintf ('.');
    IM {1} = im2vec (image);
    IM {2} = im2vec (fliplr(image));
    IM {3} = im2vec (circshift(image,1));
    IM {4} = im2vec (circshift(image,-1));
    IM {5} = im2vec (circshift(image,[0 1]));
    IM {6} = im2vec (circshift(image,[0 -1]));
    IM {7} = im2vec (circshift(fliplr(image),1));
```

```
    IM {8} = im2vec (circshift(fliplr(image),-1));
    IM {9} = im2vec (circshift(fliplr(image),[0 1]));
    IM {10} = im2vec (circshift(fliplr(image),[0 -1]));
    for i=1:10
        IMGDB {1,end+1}= string;
        IMGDB {2,end} = out_max;
        IMGDB (3,end) = {IM{i}};
    end
end
fprintf ('\nLoading non-faces ');
folder_content = dir ([non_face_folder,'*',file_ext]);
nnface = size (folder_content,1);
for k=1:nnface
    string = [non_face_folder,folder_content(k,1).name];
    image = imread(string);
    [m n] = size(image);
    if (m~=27 || n~=18)
        continue;
    end
    f=0;
    for i=1:length(IMGDB)
        if strcmp(IMGDB{1,i},string)
            f=1;
        end
    end
    if f==1
        continue;
    end
    fprintf ('.');
    IM {1} = im2vec (image);
    IM {2} = im2vec (fliplr(image));
    IM {3} = im2vec (flipud(image));
```

```
    IM {4} = im2vec (flipud(fliplr(image)));
    for i=1:4
        IMGDB {1,end+1}= string;
        IMGDB {2,end} = out_min;
        IMGDB (3,end) = {IM{i}};
    end
end
fprintf('\n');
save imgdb IMGDB;

%-------------------------------------------------------------------------------------------
% trainnet.m

function NET = trainnet(net,IMGDB)

%-------------------------------------------------------------------------------------------
net.trainFcn = 'trainscg';
net.trainParam.lr = 0.4;
net.trainParam.epochs = 400;
net.trainParam.show = 10;
net.trainParam.goal = 1e-3;
%-------------------------------------------------------------------------------------------

T{1,1} = cell2mat(IMGDB(2,:));
P{1,1} = cell2mat(IMGDB(3,:));
net = train(net,P,T);
save net net
NET = net;
```

```
%-------------------------------------------------------------------------------------------
% imscan.m
function im_out = imscan (net,im)
close all
%-------------------------------------------------------------------------------------------
% Parameters
SCAN_FOLDER = 'imscan\';
UT_FOLDER = 'imscan\under-thresh\';
TEMPLATE1 = 'template1.png';
TEMPLATE2 = 'template2.png';
Threshold = 0.5;
%-------------------------------------------------------------------------------------------
warning off;
delete ([UT_FOLDER,'*.*']);
delete ([SCAN_FOLDER,'*.*']);
mkdir (UT_FOLDER);
mkdir (SCAN_FOLDER);
[m n]=size(im);
%-------------------------------------------------------------------------------------------
% Langkah pertama
C1 = premnmx(double(im));
C2 = premnmx(double(imread (TEMPLATE1)));
C3 = premnmx(double(imread (TEMPLATE2)));
Corr_1 = double(conv2 (C1,C2,'same'));
Corr_2 = double(conv2 (C1,C3,'same'));
Cell.state = int8(imregionalmax(Corr_1) | imregionalmax(Corr_2));
Cell.state(1:13,:)=-1;
Cell.state(end-13:end,:)=-1;
Cell.state(:,1:9)=-1;
Cell.state(:,end-9:end)=-1;
Cell.net = ones(m,n)*-1;
[LUTm LUTn]= find(Cell.state == 1);
```

```
imshow(im);
hold on
plot(LUTn,LUTm,'.y');pause(0.001);
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
% Langkah kedua
while (1==1)
   [i j] = find(Cell.state==1,1);
   if isempty(i)
      break;
   end
   imcut = im(i-13:i+13,j-9:j+8);
   Cell.state(i,j) = -1;
   Cell.net(i,j) = sim(net,im2vec(imcut));
   if Cell.net(i,j) < -0.95
      for u_=i-3:i+3
         for v_=j-3:j+3
            try
               Cell.state(u_,v_)=-1;
            end
         end
      end
      plot(j,i,'.k');pause(0.001);
      continue;
   elseif Cell.net(i,j) < -1*Threshold
      plot(j,i,'.m');pause(0.001);
      continue;
   elseif Cell.net(i,j) > 0.95
      plot(j,i,'.b');pause(0.001);
      for u_=i-13:i+13
         for v_=j-9:j+9
```

```
            try
                Cell.state(u_,v_)=-1;
            end
        end
    end
elseif Cell.net(i,j) > Threshold
    plot(j,i,'.g');pause(0.001);
elseif Cell.net(i,j) < Threshold
    plot(j,i,'.r');pause(0.001);
end
for i_=-1:1
    for j_=-1:1
        m_=i+i_;
        n_=j+j_;
        if (Cell.state(m_,n_) == -1 || Cell.net(m_,n_)~=-1)
            continue;
        end
        imcut = im(m_-13:m_+13,n_-9:n_+8);
        Cell.net(m_,n_) = sim(net,im2vec(imcut));
        if Cell.net(m_,n_) > 0.95
            plot(n_,m_,'.b');pause(0.001);
            for u_=m_-13:m_+13
                for v_=n_-9:n_+9
                    try
                        Cell.state(u_,v_)=-1;
                    end
                end
            end
            continue;
        end
        if Cell.net(m_,n_) > Threshold
            Cell.state(m_,n_) = 1;
```

```
            plot(n_,m_,'.g');pause(0.001);
            imwrite(imcut,[SCAN_FOLDER,'@',int2str(m_),',',int2str(n_),'
(',int2str(fix(Cell.net(m_,n_)*100)),'%).png']);
         else
            Cell.state(m_,n_) = -1;
            %imwrite(imcut,[UT_FOLDER,'@',int2str(m_),',',int2str(n_),'
(',int2str(fix(Cell.net(m_,n_)*100)),'%).png']);
            plot(n_,m_,'.r');pause(0.001);
         end
      end
   end
end
%-------------------------------------------------------------------------------------------

%-------------------------------------------------------------------------------------------
% Third Section
hold off
figure;imshow (Cell.net,[]);
xy_ = Cell.net > Threshold;
xy_ = imregionalmax(xy_);
xy_ = imdilate (xy_,strel('disk',2,4));
[LabelMatrix,nLabel] = bwlabeln(xy_,4);
CentroidMatrix = regionprops(LabelMatrix,'centroid');
xy_ = zeros(m,n);
for i = 1:nLabel
   xy_(fix(CentroidMatrix(i).Centroid(2)),...
       fix(CentroidMatrix(i).Centroid(1))) = 1;
end
xy_ = drawrec(xy_,[27 18]);
im_out (:,:,1) = im;
im_out (:,:,2) = im;
im_out (:,:,3) = im;
```

```
for i = 1:m
   for j=1:n
      if xy_(i,j)==1
         im_out (i,j,1)=0;
         im_out (i,j,2)=255;
         im_out (i,j,3)=0;
      end
   end
end

% ------------------------------------------------------------------------------------------
% drawrec.m

function out = drawrec (in,w)

% Fungsi ini akan menggambar suatu persegi panjang 27x18
% Persegi panjang akan digambar pada tiap gambar biner yang merupakan
kandidat wajah.
% ------------------------------------------------------------------------------------------
[m n]=size(in);
[LUTm LUTn]=find(in);
out = zeros (m,n);
for i =1:size(LUTm,1)
   try
      out (LUTm(i),LUTn(i))=0;
   end
   try
      out (LUTm(i)-14:LUTm(i)+13,LUTn(i)-9)=1;
   end
   try
      out (LUTm(i)-14:LUTm(i)+13,LUTn(i)+8)=1;
   end
```

```
    try
        out (LUTm(i)-14,LUTn(i)-9:LUTn(i)+8)=1;
    end
    try
        out (LUTm(i)+13,LUTn(i)-9:LUTn(i)+8)=1;
    end
end

%-------------------------------------------------------------------------------------------
% main.m
clear all;
clc;
close all;
if ~exist('gabor.mat','file')
    fprintf ('Creating Gabor Filters ...');
    create_gabor;
end
if exist('net.mat','file')
    load net;
else
    createffnn
end
if exist('imgdb.mat','file')
    load imgdb;
else
    IMGDB = loadimages;
end
while (1==1)
    choice=menu('Face Detection',...
                'Create Database',...
                'Initialize Network',...
                'Train Network',...
```

```
            'Test on Photos',...
            'Exit');
    if (choice ==1)
        IMGDB = loadimages;
    end
    if (choice == 2)
        createffnn
    end
    if (choice == 3)
        net = trainnet(net,IMGDB);
    end
    if (choice == 4)
        [file_name file_path] = uigetfile ('*.jpg');
        if file_path ~= 0
            im = imread ([file_path,file_name]);
            try
                im = rgb2gray(im);
            end
            tic
            im_out = imscan (net,im);
            toc
            figure;imshow(im_out,'notruesize');
        end
    end
    if (choice == 5)
        clear all;
        clc;
        close all;
        return;
    end
end
```